International Education Studies; Vol. 6, No. 3; 2013
ISSN 1913-9020 E-ISSN 1913-9039
Published by Canadian Center of Science and Education

Scope Management: A Core Information System Implementation Project Pedagogy

Pierre-Majorique Léger[1], Derick Lyle[1], Gilbert Babin[1], Patrick Charland[2] & Robert Pellerin[3]

[1] HEC Montréal, Montréal, QC, Canada

[2] Université du Québec à Montréal (UQAM), Montréal, Qc, Canada

[3] École Polytechnique de Montréal, Montréal, Qc, Canada

Correspondence: Pierre-Majorique Léger, HEC Montréal, 3000, Chemin de la Côte Sainte-Catherine, Montréal, QC, H3T 2A7, Canada. Tel: 1-514-340-7013. E-mail: pierre-majorique.leger@hec.ca

Received: January 11, 2013 Accepted: January 16, 2013 Online Published: January 23, 2013

doi:10.5539/ies.v6n3p55 URL: http://dx.doi.org/10.5539/ies.v6n3p55

Abstract

This article describes an initiative to provide IS management a capstone course that builds on the zone of proximal development concept, oriented towards developing prioritization and critical reasoning skills, and to promote self-learning. Request for proposal business cases appear to offer effective mechanisms for retaining context, while constraining scope for academic purposes.

Keywords: scope management, information system, project management, enterprise ressource planning

1. Introduction

ERP systems are commercial off-the-shelf software packages that are purposely built with a wide variety of configurable functionalities meant to support the business needs of organizations (Brehm, Heinzl and Markus, 2001). The implementation of these business applications often requires IT business analysts that help the organization select the appropriate options and configure the software according to their needs. The training of these analysts requires that they are exposed to a wide variety of business processes and software solutions to develop their skills at finding and adapting the right technological solutions. Although we are talking about IT students, the focus is on business analysis, even though the end solution will involve the application of technology.

Several authors have called for the use of an integrated IT project that requires application of conceptual as well as technical skills to build this configuration competency (Seethamraju, 2007). Venkatesh (2008) suggests the use of business cases to teach students about successful contextual approaches at implementing ERP. Volkoff (2003) also suggests the use of teaching cases to introduce students to the value of best practices. In a similar manner, Draijer and Schenk (2004) suggest the use of ongoing fictional companies to train students to operate and further expand the functionalities of an ERP while Johnson et al. (2004) use a hypothetical case to illustrate the integration in an ERP system. In most instances, the case study approach allows students to develop high-order reasoning and decision-making skills through "learning by doing" (Hackney et al., 2003) which in turn increases their motivation and interest in the subject (Mustoe and Croft, 1999).

Unfortunately, ERP teaching cases are inevitably somewhat artificial. They rarely allow students to experience all the challenges of process reengineering as they do not provide students with the opportunity to interact with real employees and face management concerns (Morrell et al., 1993). As such, these approaches may be insufficient to develop the full spectrum of skills required for students to eventually lead real ERP implementation projects (Boyle and Strong, 2006).

The role of problem-based learning in the development of IT skills has not been extensively investigated. In a recent meta-analysis, Walker and Leary (2009) underline the lack of empirical evidence on the benefits of problem-based learning. Two authors in particular have called for a richer conceptualization of IT curriculum to bring in the classroom the complexity of real world problems to develop competent IT specialists (Jasperson, Carter, and Zmud, 2005; Seetemraju, 2007, 2008).

These limitations have brought researchers to propose new ways of teaching ERP solutions. Hawking et al.

(2004) called for a second wave of ERP education that goes beyond the simple demonstration of the potential of the application. They suggest more innovative and strategic use of the technology to train IS graduates on problem resolution approaches. In a similar manner, Pellerin and Hadaya (2008) stress the importance of centering the learning process on the business process redesign instead of putting all the emphasis on the ERP implementation. Léger (2006) and Leger et al. (2012) suggests a simulation game approach to provide hands-on experience with the main functionality of an ERP. Overall, their results show that students tend to learn more by applying the skills required in conducting successful process improvement projects.

When we consider ERP implementation project complexity, it is largely the business process re-engineering that drives the level of complexity, which is then propagated to the technology used to support that process, and not the other way around. It can also be argued that by spending more effort to reduce business process complexity and yet accomplish the same organizational goals; implementation cost, complexity and therefore risk are significantly reduced. In this paper, we explore the learning process of IT students. We are particularly interested in developing their ability to perform project scope management. To this end, we propose a novel course curriculum and teaching approach, that we believe will maximize their learning. We describe how the proposed approach was used in class and reflect on the observed outcomes from the experiment.

2. Information System Implementation Project Pedagogy

2.1 Vocational Learning in the Context of IS Projects

For Surendra and Denton (2009, p. 77), IT graduates must be able to "assess and understand organizational requirements and opportunities in ambiguous and messy organizational situations and design systems that accomplish requirements and leverage opportunities." The structure of traditional curriculum places emphasis on the transfer of knowledge of technology specifics, techniques and methodologies and how to use these within the context of illustrative examples. Instructors usually set a problem and demonstrate how to solve it. Then a different problem is proposed to students who need solve it the same way. Finally the theory is provided about how to manage one complex problem as well as how to deal with complexity of situations and contexts.

After hire, a typical IT analyst will go through an induction or other onboarding program to become familiar with organizational policies and procedures, and any resources such as documentation or other knowledge repositories that are available to assist them in performing their duties. After that, they are soon assigned to a team and a project, and expected to learn as they go.

The knowledge transfer of specific technologies, techniques and methods, although necessary, is not sufficient acquired learning to achieve the level of competency needed in students for them to succeed after graduation. (Debuse and Lawley, 2009; Aasheim, Li and Williams, 2009; Lee and Han, 2008). They must be able to apply this knowledge beyond limited contexts and limited complexity. There is an expectation to be able to achieve some degree of independent work. The less supervision required, and the more quickly this level matches what is typical for the hiring organization, the more competent the new hire is perceived to be. Conversely, too great a need for supervision, or lack of ability to accomplish independent work of low complexity, results in a perception of low competency.

Vygotsky (1978) introduced the concept of the zone of proximal development (ZPD) – the difference in competency an individual has when asked to perform a task on their own, compared with their competency while under the guidance and assistance of someone with greater competency. By being given tasks between these two points, learning can occur, and as a result these points shift – the individual is able to accomplish the same task with less assistance, and more complex tasks that were previously beyond their competency. The width of the ZPD can also be linked with confidence. Individuals lacking in confidence will not attempt tasks even with promised assistance, or simply require too much assistance for any learning to actually occur.

Typically associated with early childhood models of development, the ZPD concept also applies to vocational learning where guidance and assistance are synonymous with supervision, coaching and mentoring. As discussed above, there are expectations of minimum competency and how much time is willing to be spent supervising, both of which relate to the complexity of the given task. For an individual to be perceived as competent, the total range of tasks they are able to perform under varying levels of supervision must be below the upper bound of their ZPD, but above the lower bound of the level of supervision that will be tolerable to those providing assistance. Over time, they are expected to require less supervision for those same tasks and to be able to attempt more complex tasks. Therefore, the competency expected of them is that which places the complexity of the tasks assigned to them between the two points of their ZPD.

2.2 Closing the Gap: New Pedagogy for ERP Systems

The gap between their attained level of competency and that which is required is the challenge for each student to succeed at their new job. To produce students that meet industry expectations of competency at graduation requires IS curricula to accomplish at least three things: Firstly, to ensure that the tasks that they are at least able to attempt with minimal assistance are as complex as the simplest tasks they would be assigned. Secondly, that the range in complexity of tasks attemptable with supervision is wide enough to encompass the range of tasks they are likely to be faced with. This second aim is additionally important in that it also increases their chance for on-the-job learning. Finally, and most importantly of all, is the need to reconnect students with the vocational learning model itself.

This last point may not be obvious or seem particularly problematic. However, the grading structure at most universities encourages thinking that anything less than a perfect solution is a level of failure, and that the perfect solution is always both possible and attainable by the student. When faced with the complexity of the real world, this expectation can cripple the confidence of the student, resulting in a narrow ZPD. They simply do not perceive that they have any chance at solving problems of that complexity, and relegate themselves to willing hands to be given specific tasks that they can accomplish unsupervised. This is at odds with what is actually expected, and denies them the opportunity for learning.

Conversely, a perfect solution may not be well received, on account of its cost to develop, implement or operate. The Pareto Principle often applies here; arriving at a solution that addresses 80% of the problem is usually easy to accomplish, but the remaining 20% of the solution is increasingly difficult and more costly to attain. The best solution is the one with the most value for the least cost, and not the one which solves the issue the most completely.

Additionally, in universities exams and assignment questions are visited once. While there is learning from failure and reflection, the exact same problem is never revisited twice. Contrast this with the world beyond the classroom, where the re-visitation of unsolved challenges is not only possible, but often unavoidable as we must continually carry the burden of them until they are solved. In this world, a partial solution has great value, since we continually reap the rewards, can use it over time to better understand the nature of the challenge and build better solutions on top of it. But many students do not realize it is acceptable to, or know how to, divide or reframe complex problems into simplified parts and forms.

To achieve these aims, IT students must be placed in a situation where they gain experience, that is, the application of their knowledge and skills to context. This learning environment must be sufficiently close to the real context they will face after graduation (Boyle and Strong, 2006), but still within their reach in order to transfer their learning into practice. It is difficult and time consuming to create such environments and find such exercises, and both may be beyond the reach of the teacher, depending on their own level of experience with real world complexity.

If we can frame exercises in curriculum that will bring students to this awareness – that partial solutions have great value – we can preserve the complexity of reality when we bring it into the classroom. We need to develop the critical reasoning and self-sufficiency skills of students that will allow them to adapt to new contexts, and work through complexity. Given the pace of technological innovation, and the challenge for organizations to find, adopt and exploit competitive advantage at ever increasing rates, it is this very adaptability that is essential to the long term career prospects of every IT analyst.

We must set them more than one problem; firmly embedded in situational complexity as well, together to create project complexity such that the solution is no longer definite or clear. A trade-off assessment has to be made – value vs. feasibility and risk. In other words, the project must be made more difficult; not by adding multiple independent problems; but by increasing the interconnectedness between component problems, the related technology used to devise solutions, and the situational context to increase overall complexity.

The challenge is that putting all these elements together in an experience that is complex and rich is by nature unstructured. But enough structure is required to guarantee learning and a way to measure (grade) the learning that has actually occurred. Additionally, the demand on the student's time must be constrained. A problem that is too complex to resolve in the time allotted, or too difficult to dissect or simplify, is likely to demotivate students. To bring relevance to the learning, it is advisable to use current commercial technologies but we must consider that using systems such as ERP systems, can become overwhelming (Fedorowicz, Gelinas, Usoff and Hachey, 2004), particularly for students that lack exposure or training to technology of that nature.

To accomplish this, for students with a range of backgrounds and pre-requisite training, we need a mechanism

that allows for a wide range in the complexity of the solutions that students might develop given the same business case, rather than trying to develop different business cases of varying complexity. In doing so, we seek to find the level of complexity within the reach of each student to facilitate learning and maintain the original context to enable them to transfer that learning to other similar contexts. With both of these, we advance their competency and promote their confidence, which hopefully widens their ZPD.

2.3 IS Project Scope Management

Requirements or scope management is often presented simplistically in project management theory. Something is required or it isn't – it is simply either "in" or "out" of scope. As per the previous discussion however, we are able to create, and place value on the creation of partial solutions. This hidden aspect of project scope is what makes requirements elicitation and scope management critical success factors to projects. Poor requirements elicitation can be framed as an underestimation of the complexity of the business need, or the unconsidered possibility of partial value solutions. As the finer details of business needs are uncovered, scope management, or more accurately, scope complexity management is essential in order to maintain perspective and priority, and prevent the project falling into panicked chaos. In essence, scope management seeks to find that level of complexity in the solution to any given business need that yields the most gain, but is reasonably achievable for the time, cost and risk that the project sponsors are willing to bear.

It is for this reason that mature requirements management processes focus on ascertaining underlying business needs, and move away from stating one particular solution as the requirement. If the stated solution proves complex, you are left with the in/out-of scope choice, neither of which is appealing – the former due to increased cost and/or risk; the latter due to failure to meet the objectives of the project.

This ability to adjust the complexity of the problem and solution is realistic. There is typically an inverse or diminishing return between the cost and the value of the developed solution, not only in terms of the cost of development and implementation, but also in terms of ongoing operational costs. There is a natural limit to scope complexity in that the marginal value of more complete solutions at some point are less than the cost to develop and operate them. Coupled with the traditional scope constraint of limited resources to complete the project, setting the project scope is no longer a trivial decision. In fact, finding creative mid-point solutions with excellent cost-benefit trade-offs can be the most value an IT analyst can contribute to a project.

There are two other aspects of scope complexity in addition to business complexity. Technical complexity, also sometimes referred to as feasibility, plays a large role in any IT project. For any solution, there is a question of fit of any and all the technology choices available to support that solution. The higher the business complexity of the solution desired, the more sophisticated the technology required, either to make the complexity manageable, or to achieve efficiency, or both. As solutions become more complex, the ability to adapt existing software solutions becomes more limiting, and the testing effort increases. The skill of the project team must also be considered. How long will it take the implementation team to master any of the proposed technologies, based on their existing skills and prior adoption experiences? What is the risk to schedule slippage and technology failure if the technology and its application to the problem domain are new to the team? Might it be better to use a technology less suited to supporting the solution, but more familiar or "proven" to the implementation team, so as to reduce project risk?

The final aspect of scope complexity is scope interdependency. For a complex end-to-end business problem, there may be many parts to the solution. Data may have to be captured first, analyzed, and then made transparent for decisions further down the value chain. Accuracy and timeliness affect the complexity and value of the solution. These parts are connected, without one there is no complete solution. This is where traditional in/out scope thinking can fail. A project that is defined in its requirements too ambitiously has a high risk of running over time and over budget, if defined in such a way that none of the scope can be removed on account of interdependency.

But this is an artificial limitation. The ability to say: "we can build a partial solution, we can start somewhere and learn from that, build upon that," is the essence of innovation. We should ensure that projects allow for this scaling of solutions, that the project management process and organizational culture focus on business need, value partial success, and more importantly that the project team see it as a possibility.

To visualize this concept, we can think of a tree. The trunk supports the branches which hold the leaves. Even if we know that the ultimate goal is to have a complete the tree, we can begin by concentrating on the trunk and then main branches first. Even though not perfect, it is still recognizable as a tree, and potentially a solid basis for further work to have a better tree.

Bringing all three of these scope complexity aspects together, we find they are loosely related. Increased solution complexity requires more complex technology. Increased interconnectedness of "in" scope functionality requires increased solution complexity, or more sophisticated technology to mitigate and manage interrelation. With these three aspects affecting and mutually constraining scope complexity, we adopt the term "scope cube" as an appropriate visualization of the true scope management question in play for non-trivial IT projects (Fig. 1).

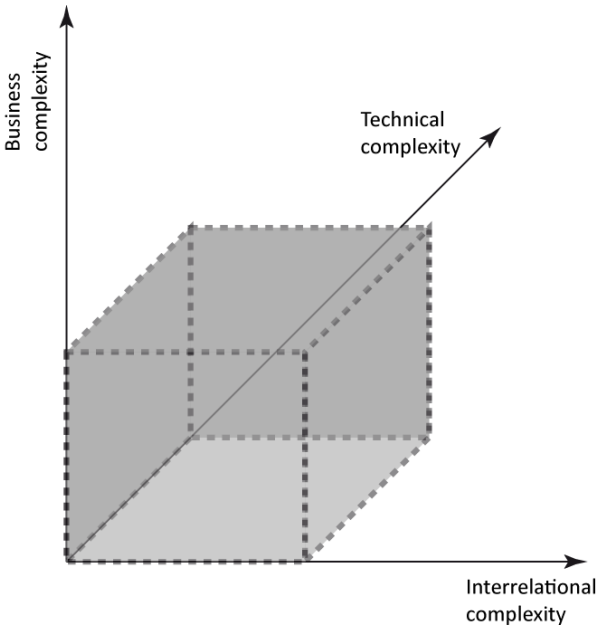

Figure 1. Scope Cube

Finally, no understanding of the challenges of adapting IT solutions to meet complex business needs can be completed without experiencing the full cycle of an implementation project. The impact of poor planning, decisions based on limited information, and poor requirements elicitation is not experienced if the project is only experienced on paper, or never beyond the planning phase. Risk assessment and mitigation is essential to the success of any project, and linked to scope ambition, not only in the planning phase but in responding to actual unfavourable eventualities along the way.

These aspects of scope management make it an important topic for the education of competent business analysts, who are the curators of the conversation between business and technology. This is also an excellent vehicle for the framing of the type of complexity exposure exercise we are looking for. The challenge is to find a context where the scope of the project is completely flexible, so that the full range of possibilities can be experienced positively; i.e., as reality meets ambition, scope adjustments are seen as necessary and realistic and not as failure. We want to allow for a lot of ambition, and yet still reward the simple, most minimal solution possible as a success and not failure.

3. Proposed Curriculum

3.1 Objectives of the Proposed Curriculum

In our approach, we aim to create a hierarchy of complexity, as opposed to a breadth of complexity. That is to say, the solutions formulated to one part of the problem impact or limit the complexity of solutions to the rest of the problem. With this problem scope, we create the interesting dynamic of scope management that is beyond "in" or "out" of scope decision-making. The scale of the original business problem, the complexity of solution, can be adjusted to fit the business needs. Or more specifically the level of business value created by the solution can be targeted.

We create enough complexity such that the perfect solution is not immediately apparent or more likely not unattainable at all by most if not all of the students. By tapping into typical student expectations of perfect solutions, this places their perceived task as well beyond their ZPD. This objective serves two purposes. To highlight the need for problem dissection and simplification, and later as a confidence booster when students

realize they are able to accomplish something valuable in contrast with an assignment that they initially thought impossible.

The emphasis on the exercise is for students to learn the importance of compromise and prioritization, how to assess business value delivered from IT, and the process of solution emergence that comes from critical thinking, trial and error, and self-learning. Acquisition of specific solutions for specific problems with specific technologies is de-emphasised as the goal. These are only useful if they can be adapted into more effective solutions, similar problems, and different technologies.

A final objective is that the pedagogy must allow for the guidance and assistance element of the ZPD concept, typically from the instructor. Opening up this role to teammates however could be especially enriching, as it's a closer match to work environments. More experienced coworkers, not supervisors, often play key roles in getting new recruits up to speed, and yet also have less tolerance for the extent and duration for which they are required to give assistance. Given different backgrounds and aptitudes between students, we don't think this is unrealistic.

3.2 Request for Proposal (RFP) with Proof of Concept

Building upon the suggestion of Seyed-Abbassi, King, and Wiseman. (2007), we suggest a real-world business project, realistic business scenario, and collaboration between business and academia. This allows for a lot of scope, in terms of complexity as well as breadth even to the point of customer "wish list" distractions. We suggest this be a team exercise, but even so, students obviously cannot be expected to actually deliver a project of this scope.

To retain both context and scope complexity, we reframe the deliverable for the project as a proposal, instead of an actual working solution. Students must formulate a proposal, and "sell" it on paper, however to keep it honest and not have it turn into an ambitious "slideware" exercise, part of the proposal is to deliver a partially functioning prototype. This proof of concept may be barely functional, but is meant to highlight the key aspects of the proposed solution, and act as a check and balance to viability. It is a competitive experience. Each team takes the role of a 3rd party integrator, and only one will "win" the contract based on the strength of their proposal. There is no formal prize or any tangible benefit, but for the purpose of completing the experience, a clear winner is defined.

Since the complete scope does not have to be delivered, only understood, the student workload is constrained and the situational complexity retained. Additionally, for the functional prototype, this is where we can have the most fun. Here, scope is completely within the control of the students. They can be as ambitious as they like, or as conservative as they like – even to the point of screenshot mock-ups only.

It also allows us to meet our objective of providing the guidance and assistance element. The instructor may coach the students into adjusting what is in-scope, and help them prioritize various elements of the final solution. They may also provide them key technical assistance to get over particular hurdles, but suggestions on where to find the answer, or how to experiment and find it for themselves, are likely to be better linked to the objectives of self-sufficiency. Most importantly the instructor is likely to need to intervene regularly to provide guidance on what are acceptable dissections and simplifications of the overall problem, and what are not. Team members with strong business acumen also have potential to fill this role, while those with stronger technical skills can provide guidance with regard to what is feasible to implement given resource constraints.

To remove focus from the completeness of the prototype, and place more emphasis on the overall approach and proposal, we suggest that grading mechanisms be adjusted accordingly. This is to downplay feelings of failure and hopelessness from the students but encourage them to try. It is also an effective mechanism for prioritization conversations that reflect the real world. Students often link effort to grades, which is in opposition to the grading weightings we suggest. However, in typical RFP situations, a lot of work is expended on proposals that are ultimately won by only one vendor. The message is to prioritize and build enough of the right pieces of the overall solution to show enough value, but not so much as to incur great expense for a solution that the customer may not agree with. This is completely in line with the learning objectives of the exercise.

With this framing, and creating a representative scope cube for the project, we are not teaching how to solve a business problem separate from that of how to then resolve the proposed business solution with technology. It is about learning to make the tradeoffs between value, cost, and risk together. More specifically, it is about the lesson of simplifying the level of sophistication of the business problem that we choose to solve, in order that the solution is feasible given the technology that is available, and time and budget constraints. This is the competency that businesses are looking for – to formulate solutions that create value and are achievable within environments filled with complexity and uncertainty.

4. A Course Experiment

4.1 Context

The experiment was performed in a capstone course in the undergraduate program for business students specializing in IT. A 60 page request for proposal was prepared to describe the business problem, including all the elements described above. Some support was provided by the collaborating enterprise in the creation of the request for proposal – clarification questions, initial proposal verification, final proof of concept, and proposal judging.

4.2 Project Description

The essence of our RFP problem was related to cost visibility and allocation in a manufacturing enterprise. Each item manufactured by the enterprise has an associated variable production cost, which varies from one production batch to another. The client needed to carry these costs over to a pricing model, such that they could ensure those costs were reclaimed and they made a profit. The problem complexity in this case dealt with the need to capture in an efficient manner costs as they occur, and carry or allocate those costs through the value chain and input them into the pricing decision. The situational complexity came from the distributed nature of the business. Any part of any production run could be sent to any warehouse, and at the time of sale, the stock in a warehouse could have come from any number of production runs, or even different factories. And yet, the customer wanted a single, simple price quotation. To resolve this problem, a solution for accurate cost capture had to be developed, along with a solution for cost allocation or identification along the distribution chain. The complexity of the second solution was constrained by the timeliness and level of granularity of the proposed cost capture solution. There was no decision to put either part of the problem in or out of scope. Both need to be included on some level in order to have any kind of solution. Limited scope would be to capture all production costs together, irrespective of location, and simply use the global average cost in the pricing decision. Scope complexity could be increased by capturing costs and calculating an average per factory, and then calculating an average cost per warehouse based on the history of which plants supplied which warehouse. Full scope would capture and propagate the exact cost history of every item in inventory right up until the point of delivery for a sale.

To address the third aspect of scope complexity, technical complexity, we made available a choice of technologies to be used to deploy the solution. Some were better adapted to certain solutions than others. The complexity of cost capture and allocation possible in the underlying operational ERP system had to be discovered. If the chosen technology already supported the need, the cost of the solution would be less than if a customization would need to be made. Students had to find these limits and make the value/cost assessment in order to make their proposal.

The best practices provided by the software editor are an excellent resource to explore different solution opportunities, because they are comprehensive and extensive, but being best practices, are not guaranteed to be complete solutions in most contexts. Students had to adapt, assemble and extend more than one best practice into their final solution. Additionally, the best practices offered ways to form part of the solution involving cost capture and business process re-configuration, but a completely different set of technologies had to be used to build the final end-user experience.

For this last part, various technical alternatives were briefly presented and teams had to choose which of them to use to complete their solution. Enough information was given for students to assess viability/feasibility given their own skills and competencies, but not enough to complete their solutions. Self-learning in their chosen technical solution was required. After identifying and understanding the business problems, students needed to evaluate the available technologies and associated best practice applications for degree of usefulness in solving those business problems. The pedagogical objective here was to learn how to explore and match many possible technical solution sets to the solving (or partial solving) of specific business problems.

At the end of the semester, students had to present their solution to a panel of reviewers from the collaborating enterprise. Feedback was provided with respect to the level of understanding of the problem communicated; appropriateness of the level of complexity tackled; and the image of competency they were able to create by the content and style of their presentation.

5. Discussion

5.1 The Challenges of Scope Management

In the RFP framing, all scope dimensions are under complete control of the teams. There was no mandated minimum set of requirements, only to have “something working” and a motivation to have better solutions than

the competition. During the semester, another scope dimension not traditionally thought of as "scope" was discovered – the quality of the implemented solution. Since a prototype was all that was being developed, the level of quality is completely optional. In real projects, this is less the case, but quality is easily sacrificed when faced with schedule and budget constraints and traditional scope (requirements) cannot be reduced. To extend the scope cube analogy, quality can be thought of as cube density. Given a cube of particular scope, resources and time expended on the project fill the cube uniformly. Fewer resources spent results in reduced quality for the same scope, or a less "solid" cube.

The reaction of most of the students to adjusting solution complexity and sacrificing quality was that it was "cheating." Faced with no other options, and some assurance that it was a perfectly appropriate response to a project in jeopardy, particularly in the RFP context, they all embraced this thinking. From there, the challenge for the students was to identify the component parts of the various business problems, and their interconnectedness. Here, true learning about understanding business requirements can occur. Creativity in finding multiple solutions at different complexity levels to the same business problems was difficult for the students, and significant coaching was necessary. Once guided however, spirited discussion about how to prioritize these, and formulate risk assessment and mitigation strategies was observed.

Finally, the greatest challenge was then to propose and convince the customer that their chosen scope and prioritization had sufficient business value. This appeared to be an entirely new world of discovery for most of the students, on which they frequently commented on aspect of the learning that they had experienced by semester end.

5.2 Selling the Solution to the Customer

Students discovered that it was not about finding the right answer, as they were typically accustomed to, but about choosing a solution that would be feasible and "sellable" to the customer – that is, choosing the level of sophistication that generates enough perceived business value with project cost that the customer is willing to accept. The students also learned that defence and elaboration of choices made is an important factor in generating client confidence and the perception of credibility to execute the proposed project successfully within budget and schedule.

There was a range of business problems, scope for the adjustment of complexity willing to be addressed within those business problems and a range of technical alternatives that could be used to solve one or more of each of those business problems (many-to-many relationship between business problem and technology). This allowed a rich selection for the technology dimension of the scope cube, and every team proposed a different technology set. Again, the need to propose and to convince the customer that their chosen technology was best and came without significant risk that could not be avoided or mitigated, was something new for the students. Many commented on the reflection that the most sophisticated or advanced technology should not automatically be chosen if you cannot demonstrate its value for its cost and associated risks.

Some business problems were not solvable with any of the technologies (business complexity high); some were solvable with only some of the technologies (technical complexity varies in context); some were limited by technical feasibility (business complexity reduced by assumption in order to meet technical feasibility). This was where the real learning about scope management and business/technology/risk tradeoffs was aimed. The firsthand experience of ambition and subsequent confrontation with reality was a rich experience for all.

Communication was another required competency – to articulate the comprehension of the business problems as well as clearly describe the proposed solution and its merits and shortcomings. Different styles of communication, for different audiences were also covered. A written proposal, a live technical demonstration, and an oral summation were all mandated.

5.3 Working in Teams

Working in teams created an additional complexity – communication, consensus (especially repeat consensus as scope decisions were constantly revisited), conflict management, resource management, and skills assessment were important aspects that each team had to deal with. Of course, the traditional concept of applying methodologies and techniques they have already learned into solving complex business problems were still an important factor. In this context, they had to learn how to find consensus to match the problem with the right methodology to resolve the problem. The competitive element created both excitement and anxiety for the students. Convinced that there were real consequences to their choices, some healthy debates about trade-offs were observed. This generated an incredible level of emotional reactions, both positive and negative, with many switches between the two.

5.4 Proposed Curriculum

To summarize this approach, the following table presents the week-by-week curriculum used in this course. This is a standard AACSB curriculum over a period of 12 weeks (3 hours weekly). The course is organized in four phases. In Phase 1 (Week 1 to Week 3), the students discover the RFP. They are assigned to teams, and must prepare an interview with the customer. On Week 3, they conduct a business requirements analysis interview with the real customer. For many of them, this is their first experience at eliciting business needs in a live format (compared with the traditional case-based approach used widely in IT education).

In Phase 2, students must prepare to present a first proposal to the customer. During this three week phase, they elaborate their solution, and prepare a convincing presentation to the customer. After presentating, the customer provides their feedback. In many cases, this is an opportunity for the students to clarify their understanding of the mandate.

In Phase 3, building on the customer's comments, the teams must prepare an answer to the RFP as well as a working prototype. Student teams are time-constrained as they have three weeks to achieve this delivery. This is where they must use their judgement, and make appropriate scoping choices. In our experience, students need the most coaching and guidance during this period. As they get closer to the deadline, the decision to reduce the scope of the working prototype gets more and more difficult. Students must also prepare a convincing document and refined presentation for final presentation on Week 11.

Phase 4 is the project post-mortem, which is performed on Week 12. We debrief the students about their experience. At this point, the competitive nature of the project is irrelevant. Students can talk openly about the problems they have been facing and can also propose ideas to improve this course.

Table 1. Course Curriculum

	Week	Title	Brief description
Phase 1	1	Introduction	The RFP is presented to the students. "Balanced teams" are put together, taking into account the technical and communication skills of students.
	2	Preparing the customer visit	In preparation for the customer visit, teams analyse the AS-IS described in the RFP. Teams must prepare a list of questions for the client to further elicit the business problem.
	3	Customer visit	The "real" customer comes to visit the student teams. Each team is allowed a fixed amount of time in order to interview the customer representative. At the end of the class, the students should be able to validate their understanding of the mandate with the customer.
Phase 2	4 5 6	Business blueprint	Over the next three weeks, students develop the blueprint of a TO-BE solution for the customer. Students are given access to SAP best practices (www.sap.com/bestpractices) to build their solutions on top of the existing technology.
	7	Customer validation	Students present their proposed blueprint to the "real" customer. This meeting provides students with the opportunity to get initial feedback from the customer relevant to the proposed solution.
Phase 3	8 9 10	Final testing	Over the next three weeks, students must develop a working "prototype" based on their proposed solution. Given the time constraint, teams must make scoping choices needed to meet the final presentation deadline.
	11	Final presentation	A final presentation is organized with the "real" customer. During this presentation, each team must convince the customer that they have the most effective solution to answer the RFP During this presentation, students must use their communication skills to convince the customer. They must also demonstrate their prototype.
Phase 4	12	Post mortem	The last week of the course is used to do a post mortem of the project and the course.

6. Conclusion

It was aimed, in this paper, to explore the learning process of IT students while being interested in developing their ability to perform project scope management. We presented and discussed a novel course curriculum, implying a coherent teaching approach which we believe can maximize students learning in project management training. We described how the proposed approach was used in class, and analyzed the observed outcome from our limited experience.

We believe that the framing with the RFP is a very effective vehicle – it allows retention of the richness and complexity of the business problems, while constraining the scope for academic purposes. There is no failure if the solution is too simplistic or not robust. What is important for students is to sell the value of their solution and their ability to execute the proposed project. The competitive nature of the RFP also brings excitement and stress of a kind often unexplored in academia but typical once they enter industry.

Our future intention is to extend and replicate this approach for additional courses. On-going research aims to gather empirical evidence in order to measure the effectiveness of our training strategy. Investigation of industry expectations of new hires, and degree of fit, between the ZPD guidance and assistance concept with that of supervision and mentoring, could also provide valuable insight.

References

Aasheim, C., Li, L., & Williams, S. (2009). Knowledge and skill requirements for entry-level information technology workers: A comparison of industry and academia. *Journal of Information Systems Education*, *20*(3), 349-356.

Boyle, T. A., & Strong, S. E. (2006). Skill requirements of ERP graduates. *Journal of Information Systems Education*, *17*(4), 403-413.

Brehm, L., Heinzl, A., & Markus, M. (2001). Tailoring ERP systems: A spectrum of choices and their implications. In Proceedings of the 34th Annual Hawaii international Conference on System Sciences (HICSS-34), January 03–06, Washington, DC: HICSS, *IEEE Computer Society, 8.*

Debuse, J., & Lawley, M. (2009). Desirable ICT graduate attributes: Theory vs. practice. *Journal of Information Systems Education*, *20*(3), 313-323.

Draijer, C., & Schenk, D. J. (2004). Best practices of business simulation with SAP R/3. *Journal of Information Systems Education*, *15*(3), 261-265.

Fedorowicz, J., Gelinas, U. J. Jr., Usoff, C., & Hachey, G. (2004). Twelve tips for successfully integrating enterprise systems across the curriculum. *Journal of Information Systems Education*, *15*(3), 235-244.

Hackney, R., McMaster, T., & Harris, A. (2003). Using cases as a teaching tool in IS education. *Journal of Information Systems Education*, *14*(3), 229-234.

Hawking, P., McCarthy, B., & Stein, A. (2004). Second wave ERP education. *Journal of Information Systems Education*, *15*(3), 327-337.

Jasperson, J. S., Carter, P. E., & Zmud, R. W. (2005). A comprehensive conceptualization of post-adoptive behaviors associated with information technology enabled work systems. *MIS Quarterly*, *29*(3), 525-557.

Johnson, T., Lorents, A. C., Morgan, J., & Ozmun, J. (2004). A customized ERP/SAP model for business curriculum integration. *Journal of Information Systems Education*, *15*(3), 245-253.

Lee, C., & Han, H. (2008). Analysis of skills requirement for entry-level programmer/analysts in Fortune 500 Corporations. *Journal of Information Systems Education, 19*(1), 17-28.

Léger, P. M. (2006). Using a simulation game approach to teach enterprise resource planning concepts. *Journal of Information Systems Education*, *17*(4), 441-447.

Léger, P. M., Charland, P., Pellerin, R., Babin, G., Robert, J., & Cronan, T. P. (2012). Authentic OM problem solving in ERP context. *International Journal of Production and Operations Management*, *32*(12), 1375 – 1394. http://dx.doi.org/10.1108/01443571211284151

Morrell, J. S., Freeman, J. L., Serrano, F., & Mock, R. (1993). Hands-on experience for students in information systems design. *Journal of Applied Business Research*, *9*(4), 141-145.

Mustoe, L. R., & Croft, A. C. (1999). Motivating engineering students by using modern case studies. *European Journal of Engineering Education*, *15*(6), 469-476.

Pellerin, R., & Hadaya, P. (2008). Proposing a new framework and an innovative approach to teaching

reengineering and ERP implementation concepts. *Journal of Information Systems Education*, *19*(1), 65-74.

Peslak, A. R. (2005). A twelve-step, multiple course approach to teaching enterprise resource planning. *Journal of Information Systems Education*, *16*(2), 147-156.

Seethamraju, R. (2007). Enterprise systems (ES) software in business school curriculum - evaluation of design and delivery. *Journal of Information Systems Education*, *18*(1), 69-83.

Seethamraju, R. (2008). Enhancing student learning of business process orientation and enterprise integration through ERPsim game. *ERPsim User Group Seminar*, April 17-18, Montréal.

Seyed-Abbassi, B., King, R., & Wiseman, E. (2007). The development of a teaching strategy for implementing a real-world business project into database courses. *Journal of Information Systems Education*, *18*(3), 337-343.

Surendra, N., & Denton, J. (2009). Designing IS curricula for practical relevance: Applying baseball's "Moneyball" theory. *Journal of Information Systems Education*, *20*(1), 77-86.

Venkatesh, V. (2008). "One-size-does-not-fit-all": Teaching MBA students different ERP implementation strategies. *Journal of Information Systems Education*, *19*(2), 141-146.

Volkoff, O. (2003). Configuring an ERP system: Introducing best practices or hampering flexibility? *Journal of Information Systems Education*, *14*(3), 319-324.

Vygotsky, L. S. (1978). *Mind and society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Walker, A., & Leary, H. (2009). A problem based learning meta analysis: Differences across problem types, implementation types, disciplines, and assessment. *The Interdisciplinary Journal of Problem-based Learning*, *3*(1), 12-43. http://dx.doi.org/10.7771/1541-5015.1061